 doi:10.20944/preprints202202.0106.v1

AccessChain: An Access Control framework to protect data access in blockchain enabled Supply chain.

Aaliya Sarfaraz*, Ripon K. Chakrabortty, Daryl L. Essam

School of Engineering and Information Technology, University of New South Wales, Canberra 2600, Australia

Abstract

In recent years supply chains have evolved into huge ecosystems, demanding trust, provenance, and data privacy. Since blockchain technology (BCT) allows for the development of a distributed environment, it is ideal for supply chain management (SCM) applications. However, concerns regarding data privacy have impeded the development of blockchains. Despite the fact that some blockchains can restrict participants to read and write data, the blockchain's transparency makes protecting sensitive data challenging. To solve the data privacy challenge, this paper proposes a framework, *AccessChain*, that is an SCM access control framework based on an attribute-based access control (ABAC) model that restricts access to competing parties while allowing for network scalability. This proposed AccessChain model has two types of ledgers in the system: local and global. Local ledgers are used to store business contracts between stakeholders and the ABAC model management, whereas the global ledger is used to record transaction data. *AccessChain* can enable decentralized, fine-grained and dynamic access control management in SCM when combined with the ABAC model and BCT. This paper's experimental results illustrate that high throughput can be achieved in a large-scale request environment while maintaining data privacy and sustaining a scalable network.

Keywords: Blockchain, Supply chain, Access control, ABAC, Data Security and Privacy,

1. Introduction

A blockchain is defined as a series of blocks that hold tamper-proof data transactions. Nakamoto [1] first proposed blockchain as a way to store and share Bitcoin transactions. Each blockchain offers a decentralized information exchange without the need for a mediator. Apart from digital currencies, blockchain can be utilized in a variety of supply chain management (SCM) applications [2, 3], and in doing so have since established a new paradigm for supply chain data integrity and transparency. The decentralized framework of blockchain can be used to provide reliable data transmission for SCM. Supply chain operations generally include several stakeholders (e.g., suppliers, manufacturers, third-parties, retailers), therefore, having a transparent operational immutable ledger may be quite beneficial. By introducing blockchain technology (BCT) into an SCM, stakeholders can gain competitive advantages by enhanced data visibility, automated purchasing and payment processes, lower risk of errors, and protecting the SC against counterfeiting [4].

The data on a blockchain can be divided into two categories: user identification and transaction records. However, transactional data (i.e., manufacturing records, suppliers information, and consumer demand data) are valuable assets for any SCM, and thus, their encryption or privacy is vital [5]. It is established that on a public ledger, transaction data is accessible to all participants, whereas on a private ledger; read and write permission is determined by a permissioned blockchain [6]. Participants who may share resources and have reading and writing privileges can be restricted by using a consortium blockchain [7]. Despite the fact that these features have increased the technological acceptability of blockchains in the SC sector, data privacy remains a concern. In some cases, regardless of the fact that access to the

*Corresponding author
Email address: a.sarfaraz@student.adfa.edu.au (Aaliya Sarfaraz)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

ledger is restricted, anyone that matches the accessibility criteria can still access the data from the ledger [8]. Thus, if data is shared with a huge number of participants without data privacy, businesses' objectives may be jeopardized [9]. Limiting the network to members that are part of a specific production structure is an option, but this limits the SC's flexibility. Therefore, it's vital to protect data privacy by employing a fine-grained access control framework when transferring business data such as locations, manufacturing materials and demand data. To enhance data sharing in SC, a balance between data accessibility and data privacy is essential [10]. As it is evident that the possibility of data breaches makes participants more reluctant to share personal information. In addition, inappropriate data sharing can also cost a business a lot of money in the form of fines for privacy invasion [11].
However, since data is exchanged across several stakeholders, finding this balance is extremely challenging. Such businesses are diversified, which means they operate in a variety of ways and employ a variety of data models. To make things work, they must first agree on who has access to their data and how they can trust one another before publishing it. There has been a significant amount of research on supply chain data privacy [12, 13, 14]. For example, Tian [15] addresses privacy concerns in a proposal for food safety monitoring by solely saving "key information" on the distributed ledger and outsourcing storage of comprehensive data to trusted third parties. Ferdousi et al. [16] proposed a distributed ledger that provides pseudonymity. However, due to persistent user IDs and a one-to-one mapping of business operations to publicly accessible transactions, their approach is vulnerable to correlation. Data ownership and the privacy of sensitive data are issues that must be addressed, therefore data access is a crucial concern. The subject of access control is important in SCM, along with other security concerns [17]. Access control is a critical resource restriction tool that has been widely used in a variety of applications, such as in IoT [18], healthcare data [19] and cloud Computing [20]. Access control can be considered as a kind of security that ensures that only permitted businesses with access control policies can access the required information. Typically, a robust access control framework covers three major security concerns: Accountability, Authorization, and Authentication [21]. Access can be enforced through many types of access control models: discretionary access control (DAC), mandatory access control (MAC) and Role Based Access Control (RBAC), are examples of traditional access control models [22]. These models can give fine-grained access control over resources. However, their nature is extremely centralized, with the drawbacks of single-point failure, difficulty in scaling, and low throughput for large-scale dynamic frameworks [23]. It is challenging to fulfill the AC needs in an SCM framework with centralized AC. Section 3 further explores these models. To overcome these challenges, attribute-based access control (ABAC) is used to enforce access restrictions based on the attributes of the subject, resource, action, and environment involved in an access event. ABAC, also known as policy-based access control, first separates the user, resource, permission, and environment attributes, then fuses their relationships, and eventually converts permission management into attribute management, resulting in a fine-grained and dynamic access control framework [24].

Despite blockchains having several advantages, scalability remains a major bottleneck when it comes to implementing one in a supply chain setting. Yet, it seems to be critical for blockchain acceptance in large-scale networks with growing number of participants, such as SCM. As a matter of fact, as the number of nodes in a network grows, the transaction volume in SC tends to also grow. Several research works have used various scaling strategies to address this scalability problem, such as: Khalid et al. [25] that proposes Blockchain Lightweight Scalable (LBS), which has been tailored for IoT applications. The suggested blockchain has created an overlay network on which devices that handle blockchain management and higher resources can achieve a distributed mechanism. The overlay network has been designed to lower basic expenses by delegating public chain maintenance to specific clusters and cluster leaders. The fundamental disadvantage of the this approach is that they do not consider the verification requirement and anonymity. Our proposed framework addresses all security challenges and objectives, such as data integrity, access control and authentication. To solve the scalability challenge, we use the sharding concept. Sharding is a method that divides a blockchain into numerous shards and allows participating nodes to execute and store transactions from only a few of the shards. Sharding has been shown to be one of the most practical ways to construct a scalable framework [26] and has lately been investigated in relation to blockchain framework [27, 28].
The key issue of the above methods is that none of them address all of the security concerns. Some of them deal with privacy issues, while others deal with the issue of dynamic access for data privacy. Furthermore, a scalable network design is essential because the transaction load in supply chains is likely to expand dramatically. To address the limitations of the above works, this paper proposes AccessChain, a data privacy-preserving framework for SCM that is based on the ABAC model. AccessChain manages two different ledgers to decouple stakeholders' business contracts

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

and business transactions. We structure the shards by geographical regions called (i) Local ledger: a public blockchain for managing business contracts and data access permission, with a number of regional SC participants and validators that are responsible for maintaining each local ledger. (ii) Global ledger: a private blockchain for logging supply chain transactions. A Validator is able to collect all business contracts and access rules for read and write access from businesses, authenticate them, and authorize them to the global blockchain. AccessChain provides dynamic access control management and addresses the scalability problem in SCM by leveraging a distributed framework. The following are the major contributions of this work:

i A multi-blockchain data privacy-preserving framework (i.e., AccessChain) is proposed that provides data privacy via fine grained access control for SCM problems.
ii To maximize scalability, two distinct ledgers, a global and a local ledger, are used to store business contracts and business operations, respectively.
iii In order to show that the framework is protected against network threats, security analysis is carried out.

The rest of this paper is organized as follows. Section 2 states the related work, and Section 3 details relevant background knowledge. In Section 4, our proposed AccessChain model is presented, and details of our experimental evaluation and threat model of the proposal is in 5. Finally, Section 6 gives the conclusion.

2. Related Work

Numerous research on blockchain technology focuses on providing an access control system, either in the context of specialized area like healthcare, IoT, or as a general access control system that may be used for a wide range of applications. In this section, we discuss recent research in all areas that aims to solve the challenge of data access management in blockchain-based distributed systems. We first review access control models, then discuss blockchain-based data access.

2.1. Access Control Models

An efficient access control system must address the most critical security concerns and focused on scalability, flexibility, and consistency factors. To solve data security concerns in distributed networks, numerous access control approaches with distinct objectives have been developed. Classical access control systems, such as RBAC, ABAC, and DAC were proposed as solutions to the problem of access control in large networks. It is worth mentioning that in both the DAC and RBAC schemes, validating subjects' access permissions is often done by a centralized authority, which can lead to a single point of failure [29]. To overcome this shortcoming, ABAC is used to limit the amount of rules, an ABAC model is made up of a set of rules that define requirements for a set of properties related to the subject, object or environment [30]. The rules are integrated and they must be satisfied in order for access permission to be given. ABAC is gaining popularity since it has the potential to combine the demonstrated benefits of DAC and RBAC while also overcoming their flaws. There have been several proposals for the ABAC model, such as the Usage Control (UCON) [31] model. UCON is attribute-based, but instead of focusing on core ABAC principles, it concentrates on advanced access control capabilities, including modifiable attributes, continuous enforcement, liabilities and restrictions.

Capability-based access control (CapBAC) is thought to be a potential option for distributed networks [32]. CapBAC-based schemes assign access rights to subjects based on the concept of capability. An access right is a transferable token of authorization that defines a set of access permissions for every subject [32]. Access Control List (ACL) and Capability are often used in access control management [33]. On the other hand, ACL is a centralized solution to enable administrative activities with improved traceability. Each object in the ACL model has an access control list that saves the subjects and their object access privileges. However ACL cannot handle complexity and is prone to system failure due to its centralized management feature. Similarly, each subject in the capability model has a capability list that specifies its access privileges to all objects. Skinner et al. [34] presented a CapAC model for implementing access control policies for an IoT network. However, the CapAC approach relied on a centralized authority and failed to consider lightweight requirements for smart devices. Furthermore, numerous models were presented to address these challenges (e.g. capability propagation and revocation [35]), such as Secure Identity-Based

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

Capability (SICAP) [35], Capability based Context-Aware Access Control (CCAAC) [36], and Distributed Capability based Access Control (DCapAC) [37]. Existing access control approaches have some drawbacks [23], since they are user-centric and ignore the organization's relationships. To address these points, the access control system should be distributed to avoid single points of failure, adaptable and scalable to handle a large number of users, dynamic, trustworthy, and must be capable of protecting the privacy, integrity, and anonymity of members of the network [38].

2.2. Blockchain-based Data Access

With the evolution of blockchain technology, services were created with the goal of facilitating and strengthening supply chains [39]. Blockchain is a distributed, transparent, traceable and immutable ledger in which blocks are added in chronological sequence [40]. However, due to the decentralized nature of blockchain, it is critical to ensure reliable access control of sensitive information. Therefore, access control is a vital mechanism for ensuring data access is not manipulated or compromised, and for also preventing unauthorized capturing [41]. Given the security concerns surrounding access control in SCM networks, blockchain technology, which is decentralized and tamper-proof, can be utilized to effectively store access control policies [42]. The idea of using blockchain to store access control policies has also recently attracted a lot of interest. Maesa et al. [43] employed blockchain technology to create and manage access tokens and allows distributed transfer access across network users. However, their approach continues to rely on an external centralized policy database to retrieve access rights based on blockchain linkages.

FairAccess [44, 45] is a blockchain-based access management framework for Internet of things (IoT) networks. FairAccess is a one-of-a-kind access token-based transaction for access control. The resource owner can set access policies and create access tokens for any peer. Additionally, by attempting to transfer a token, a token owner can delegate access to a new owner. The sender incorporates access control restrictions in the transaction output's locking scripts while transmitting a token. The receiver must first unlock the locking script to verify provenance of the token. Although using locking scripts for access control is a good option, the computational capabilities of locking scripts are restricted. Other drawbacks with FairAccess include the fact that if a token expires or is revoked, the subject must contact the owner to obtain a new token. Moreover, for this access framework, at least two blocks must be mined to the blockchain for a new token to be effective, making it expensive and time consuming to gain access. Xu et al. [46] presented a distributed ledger based access control (DL-BAC) approach for Web applications that makes choices and grants access using an access control list (ACL).

Zhang et al. [47] presented an access control framework based on smart contracts to automate access control. Liu et al. [48] proposed an access control system, based on the Hyperledger Fabric blockchain platform and ABAC model. Smart contracts were also included in the system to build access control techniques for various user attributes. Likewise, the authors of [43] proposed a framework for distributed auditability, which prevents a third party from refusing privileges granted by an enforceable policy. However, the approach continues to rely on an external centralized policy database to retrieve access rights based on blockchain linkages, and the experimental results are not presented. To tackle the scalability issue of distributed network authentication, many researchers have proposed several solutions. Zyskind et al. [49] used blockchain to record, query, and share data. Only the pointer to the data was stored on the blockchain, as the data was stored in an off-blockchain network. However, their work only facilitated the development of ACL-like rules and did not support other variables in the authentication process. Jiang et al. [50] outlined how healthcare-related data can be securely transferred and used two interconnected blockchains to regulate healthcare data. Furthermore, once the data's privacy and authenticity have been verified, the fairness-based packing technique was used to boost the framework's throughput. A similar strategy [50] was proposed for healthcare entities exchanging data, that meets both privacy and authenticity standards. Flapper [51] presented a user access control framework for supply chain visibility using smart contracts to offer trust, security and access control using the eXtensible Access Control Markup Language (XACML) standard. However, the study lacks any results, as well as a description of policy administration and smart contract deployment was missing. A distributed access control framework was proposed in [52], where transactions were stored in a SC on a public BC. However the framework relied on a set of administration nodes to operate as a hub for access control, resulting in a paradigm of centralized management.

2.3. Summary of the literature review

Based on the above observations, it can be claimed that blockchain has been investigated as a back-end design for a distributed access control framework in a number of research works. However, the majority of research that combines

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

blockchain technology and Access Control are focused on one of three fields: IoT, health care or cloud storage. The state-of-the-art retains the following research gaps: explicit access criteria for supply chain participants in terms of accessing blockchain data, and a scalable blockchain architecture that can support higher transaction volumes. To that end, to advance the literature in the supply chain area, our study addresses the aforementioned research problems and contributes to the literature. We propose an access framework for supply chain applications. This study presents *AccessChain*, a multi-blockchain data privacy framework that provides data privacy via fine grained access control by our ABAC model. Each local access point ledger in the framework provides access control policies and dynamic access right verification by verifying the subject's contract. In addition, the local access ledger will allow a user to add, update, and delete access control policies in the contract. When a subject requests access control in the global ledger, the contract is evaluated and verified, guaranteeing that access control is accurate.

3. Preliminaries

In this section, we present our supply chain use case and briefly discuss a few widely used access control models. We also provide a brief summary of the ABAC model, which serves as the foundation for our data access control.

3.1. Supply Chain Use Case

The automotive supply chain (ASC) is a complex network that comprises a wide range of automotive parts and many stakeholders, ranging from suppliers and manufacturers to vendors and customers [53]. The ability to access real-time product information is vital to an ASC's efficiency. Traditional ways of managing and regulating ASC are adequate, however blockchain could be a strong motivating factor to improve existing ASC. It can assist in the supply of accurate, real-time information to all parties involved, allowing them to monitor product status. The blockchain application can also be used to increase transparency during the process. If the entire process was transparent, it would empower automotive suppliers to build a customer-centric business model that would allow them to provide exceptional consumer value. Blockchain is already being used in the production process by companies like Ford and BMW [54]. Ford, BMW, Renault, and GM have pledged their support for the Mobility Open Blockchain Initiative (MOBI) [55]. ASC, like any other SC, is under intense pressure to improve transparency and reduce process complexity. As a result of this endeavor, ASC stakeholders require robust and dependable technology for information sharing and monitoring. However, the transparency property of blockchain makes it more difficult to protect sensitive data, as data is shared vertically and horizontally with all stakeholders. As a result, not all data can be provided because a supplier could be a competitor who could use the information to their benefit. There is a clear conflict between access control and data accessibility when sharing data in ASC [56]. In order to examine these dynamics, we chose ASC as our use case to show how our access control approach works. In our framework, it has been assumed that multiple local chains operate in parallel in each region. Each of these local chains, called *Access ledger*, are a public blockchain network with administrators in charge of registering the business contracts of participants, as shown in Figure 1. A supply chain entity must first register with their region's certification authority (CA) and get an identity, which verifies their digital profile on the ledger. In assess point ledgers, each participant's access privileges are defined using the ABAC model. Stakeholders in the ASC record business transactions on a private blockchain, known as the *global ledger*.

3.2. Access control models

Each request by a framework user to access a resource in the network should be filtered, hence access control is concerned with network access privileges [57]. Access control systems can be used in a range of areas and at various levels in software and hardware. In this section, we present the literature based argument that allows us to choose ABAC for our framework over RBAC and DAC. There are a variety of access control models to choose from, however all types of access control can be traced back to one of the three basic models: DAC, RBAC, and ABAC. To regulate how users access resources, each model employs a different set of techniques.

- **DAC:** is a type of access control that allows or limits user access depending on an access policy set by the resource's owner. DAC framework is established using user credentials such as login information [24]. DACs are discretionary since each user can provide other users access to authenticated resources or data. To put it in other words, the user determines their own resource access privileges.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

- **RBAC:** grants access to users depending on their responsibilities or roles within a network. Users are only given access to critical data [24]. Numerous characteristics, such as authority, responsibility, and job expertise, can determine access. Furthermore, access to resources might be restricted to certain operations, such as reading, writing, or updating.

- **ABAC:** offers access to users based on a collection of attributes. Permissions can be based on the user's type, location, department and other attributes, allowing for a more straightforward control structure that reflects the physical aspects of the network [58]. ABAC simplifies the expression of a comprehensive, sophisticated access control policy by examining a user's attributes information that are already known and frequently kept in a system [59].

DAC vs RBAC vs ABAC: Although DAC is easy to implement, it has significant drawbacks that make it unsuitable for use in a complex SCM environment. The main difficulty is that due to the enormous number of generated log entries, monitoring is challenging [60]. Despite the fact that RBAC is a popular choice for organizations, the RBAC technique has a number of limitations, including the inability to construct rules using parameters that are unknown to the framework [61]. Furthermore, because the RBAC paradigm is primarily focused on static organizational roles, RBAC designs provide issues with demand dynamic access control frameworks. No multi-factor decisions are supported by RBAC. The ABAC approach, on the other hand, offers significant advantages that are tailored to our needs, such as: ABAC may be automated to modify authorization, and once everything is set up, there is less overall management required [58]. When properly configured, it is also reliable. Most importantly, ABAC facilitates access control actions without the user's prior comprehension of the resource [61] and has been widely used in the literature [48, 62, 63]. RBAC and ABAC differ significantly in terms of their static versus dynamic nature. RBAC is more static and employs role-based access control, whereas ABAC is more dynamic and uses relation-based access control. ABAC is based on attributes, which can change frequently, but RBAC is based on roles, which are usually quite static inside a network. RBAC allows one to define access controls in broad strokes, whereas ABAC allows for greater refinement. An RBAC system grants access to all employees, whereas an ABAC policy only grants access to administrators in some specific department or region. Table 1 provides a comparison that takes relevant parameters into account. Based on this table, we may conclude that ABAC is the most suitable access control approach for our supply chain application, as it prioritizes scalability and flexibility. Access control for supply chain systems should be scalable, flexible, efficient, and trustworthy, and must be sufficient to secure the supply chain and its components' privacy, integrity and privacy. Usually a supply chain applications involve multiple stakeholders like manufacturer, producers, transporters, retailers, and customers, and it is critical to develop trust between these entities. When it comes to maintaining participants identities, the ABAC approach offers a lot more freedom and makes it useful for supply chain access control because it allows stakeholders to remain anonymous. Furthermore, ABAC allows supply chain managers to apply access control policies to an unlimited number of participants without having any prior knowledge of the subject. One of the biggest benefits of ABAC is that it makes it simple to add new users.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

Characteristic	Discretionary Access Control	Role Based Access Control	Attribute Based Access Control
Scalability	yes	No	Yes
Performance	low	high	high
Granularity	High	low	High
Flexibility	yes	yes	yes
Security level	low	high	high
Custom permissions	yes	No	yes

Table 1: Comparison of Access control model.

3.3. ABAC Model

ABAC is a type of logical access control that includes access control lists, role-based access control and its own method for granting access based on attribute analysis. ABAC regulates system resource access by comparing policies to user properties such as subject (user), object (resource) and environment. When making ABAC decisions, both subjects and objects have attributes, and the conditions of the environment may be taken into account. In essence, this means that it may utilize key-value combinations like Role = Production Manager and Category = Manufacturer to define rules in eXtensible Access Control Markup Language (XACML) [64].

ABAC is defined as follows:

- **Attributes** are traits of a subject, an object, or the conditions in which they exist. A name-value pair provides the information for attributes.
- **subject** is a user, it can be a human or a device, who makes requests for access to execute actions on objects. One or more qualities are given to subjects. Assume that subject and user are synonymous for the purpose of this article.
- **Object** is a system resource. It can be any requested resource, as well as anything that a subject can use to complete an operation; including data, services and devices.
- **Environment** is the context in which an access request are made. The time, date, location and current risk are examples of environment features that are independent of subject or object.

4. AccessChain framework

AccessChain, is a blockchain-based access control framework for SCM. As shown in Figure 1, it consist of two key components: access point Ledger and global Ledger. Each key component is explained below.

4.1. framework overview

In this section, we will discuss *AccessChain* framework, which uses two separate ledgers to store business contracts, access privileges and trade transactions. Figure 1 depicts the framework and relationships of Accesschain's core components.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

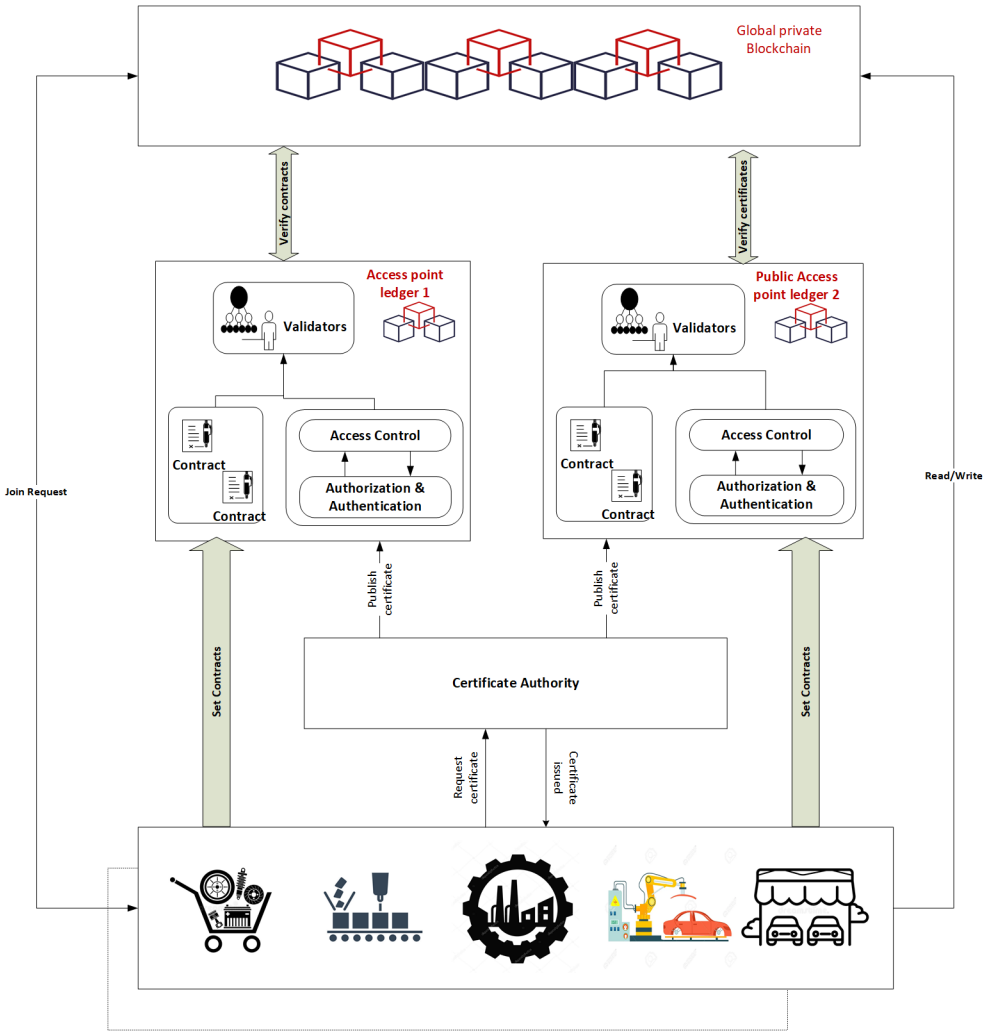

Figure 1: AccessChain framework.

Stakeholders must first register with the CA, after which CA will grant them business credentials. To place their business contracts in access point Ledger, stakeholders must first confirm that they are registered users. The stakeholders then place business contracts and assign access privileges to other stakeholders involved in the contract on access point Ledger. After they have established their identities and business contracts, the participants send a join request to the global ledger administrator. The stakeholder must authenticate their trade identities without disclosing them. The global ledger administrator validates and approves the seller's registration request in global ledger. The stakeholder can then log transactions on global ledger.

4.2. Access point ledger

Access point ledger is committed to providing business contracts and access control policies to all participants. The IDs on access point ledger can be verified by a centralized authority. In blockchain-based frameworks, public-key cryptography is frequently used to authenticate users in a network. As part of our design concept, cryptographic algorithms such as SHA256, digital signatures, and state of the art elliptic-curve cryptography (ECC) [65] are used to digitally sign transactions. In access point ledger, business contracts are created using an approach that is similar to that utilized in the real world.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

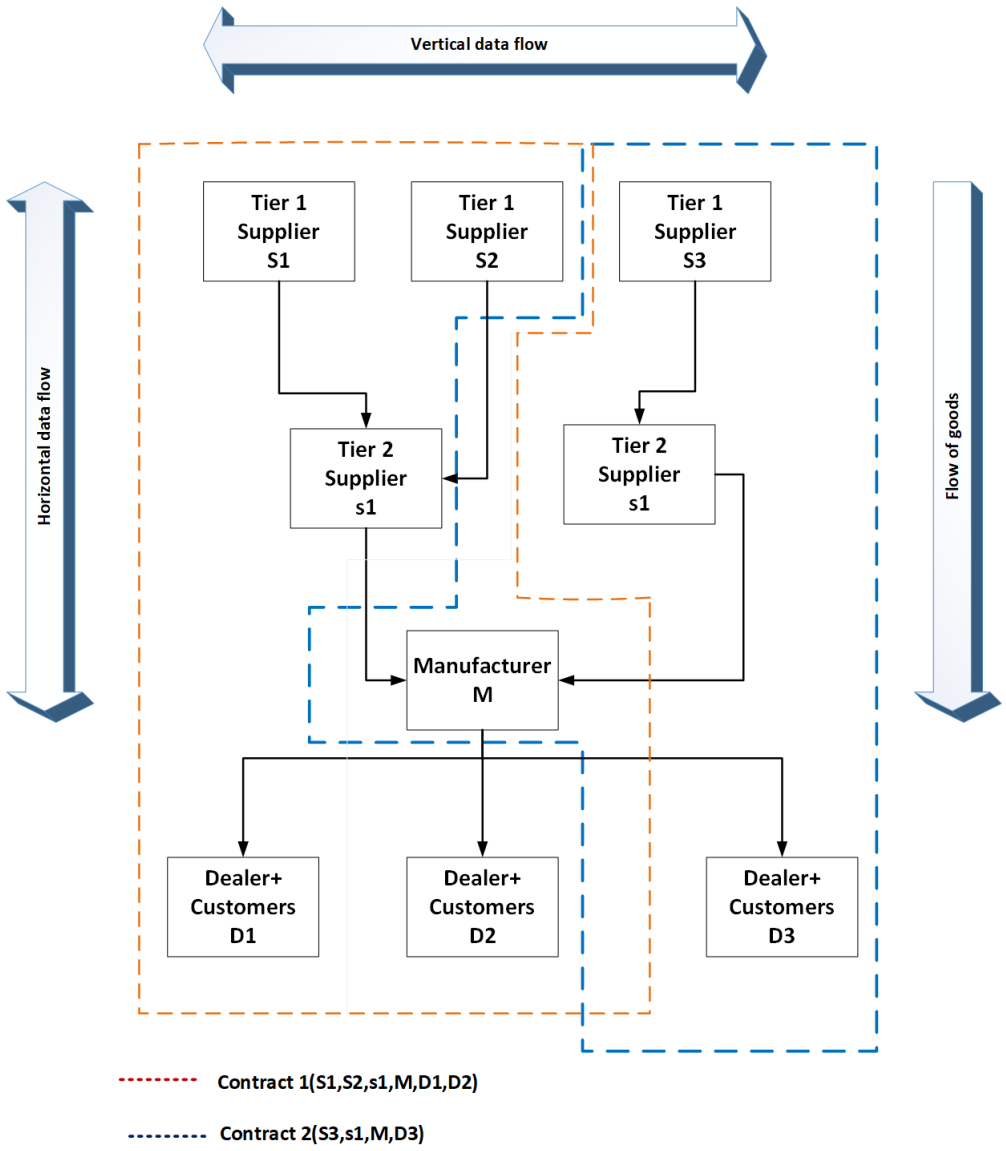

Figure 2: SC-Contract: Stakeholders in a Blockchain-based data access contract.

Consider an automotive supply chain, which is made up of several businesses. As discussed in Section 3.1, an automobile SC has multiple suppliers for various raw materials. There is no need to communicate every piece of information among several businesses. For example, it is not necessary for a leather supplier and a steel supplier to be on the same business contract. To protect data privacy and security, only the relevant entities required for a swift business transition are included in the contract. Furthermore, every contract must contain an expiration date.

$$Contract \rightarrow [Policy|ID_{\text{user}}|ExpDate|Sig_{\text{Owner}}] \quad (1)$$

where *policy* is the user access model, ID_{user} is the identifier of the participants, Sig_{Owner} is the signature of the contract owner and *ExpDate* represents the contract's expiration date. Only related businesses are included in a contract and have read/write, update/delete permissions. Figure 2 depicts a number of businesses in an automotive supply chain that are linked via contracts. There are two ongoing contracts, in *contract 1*, manufacturer has tier 1 and 2 suppliers (S1, S2), s1 along with distributor D1 and D2. The automotive Manufacturer has another *Contract 2* with other businesses

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

S3, S1 and D3. There is no contract path from S1 to D3, this indicates that S1 has no access to data from D3 and vice versa. In summary, access point Ledger stores credentials, business contracts, and access control rules that are not accessible to the general public and can only be accessed with the contract owner's, i.e. the manufacturer's, approval.

4.3. Global ledger

A global ledger is where the actual business transactions are recorded and it contains a full history of transitions. When a request to read/write or update/delete is made, validation is carried out by ensuring that the participant has the required permits and that the contract is still active before granting permission. The *ID* of the access point ledger where the contract for the access is held must be kept as supplementary data with each transaction to the global ledger.

$$Transaction \rightarrow [ID_{\text{user}}|ID_{\text{ledger}}|Sig|data] \tag{2}$$

Where ID_{user} and *Sig* is the signature and public key of the participant and $\text{ID}_{\text{ledger}}$ is the location of access point ledger. The global ledger utilizes the *RequestAccess* function to verify the permissions from the access point ledger before logging a transaction and the contract's validity is double-checked using a function known as *ContractValidation* (explained in algorithm 2). Permission is given if both functions produce a positive return. Otherwise, the access was denied through *RevokeAccess*. The main function of access control management is Algorithm 1. To begin, it obtains the attribute set specified by *GetAttribute()*. It will provide an error message if the returned result is empty, indicating that there is no contract to support the request. If the returned result is not empty, it means that at least one contract will be obtained. Finally, it calls the *policy()* function to verify that the desired user ID and permission are in the contract and returns access to the user, failing which an exception is returned, where *policy()* contains the ABAC access privileges for the specific contract.

Algorithm 1 RequestAccess(): Check user's privileges.

```
1:  procedure RequestAccess( )
2:      sub, obj, env, opr ← getAttr(ABAC)
3:      contract ← C_1, C_2, ..., C_n
4:      for all con ∈ contract do
5:          if con = null then
6:              return error
7:          else
8:              policy ← getPolicy()
9:              if policy ∈ (ID_user, permission) then
10:                 Access ← getAccess(UID)
11:                 grant ← TRUE
12:             else
13:                 grant ← FALSE
14:             end if
15:         end if
16:     end for
17:     return grant
18: end procedure
```

Additionally, the global ledger must validate the contract's validity, as stated in Algorithm 2. To meet the requirements, a legitimate contract must not be expired, and not be canceled by the administrator.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

Algorithm 2 ContractValidation(): Check contract validity.

```
procedure ContractValidation( )
    contract ← C_1, C_2, ..., C_n
    for all c ∈ contract do
        if c ∉ (ExpDate, status) then
            valid ← FALSE
        end if
    end for
    return valid
end procedure
```

4.4. Mining

Mining is the operation of adding new transactions and contracts to the ledger. Validators are the nodes responsible for this operation. Each validator creates a block of generated transactions and contracts and uses the proof-of-work consensus process to solve an incredibly tough cryptographic puzzle connected to their block until one of them succeeds. The successful validator then publishes their block to the network's other nodes for validation. Proof-of-work [1] was chosen for the access point ledger because it offers a higher level of security than other consensus methods and is the best for a public blockchain. It is worth noting that mining in the global ledger is based on the concept of proof-of-authority [66], which relies on trustworthy and recognized validators to generate blocks, rather than their own computing power. Through mining, the network creates a tamper-resistant state on the blockchain, which is essential for secure access control for the SCM network.

4.5. Workflow

This section outlines the workflow of *AccessChain*. The entire framework's workflow is represented in Figure 3. The framework's foundation step is the blockchain network's registration. Credentials for all participants should be created first before joining a blockchain network. Here, CA is the one that generates all certificates.

$$CA \rightarrow Credential_{\text{user}} \tag{3}$$

Once the stakeholders have their credentials, and they wish to write a business contract on access point ledger. They do this before logging into the global ledger. They create a business contract as well as an access control policy. This procedure involves the stakeholder determining and designing the access policy beforehand and then publishing it to the access point ledger, which is defined based on the subject (user), object (resource), operation, and environment parameters (discussed in Section 3.3). Validators are in charge of saving and managing contact information in the ledger.

$$Determine(Subject, Object, Operation, Environment) \rightarrow ABAC \tag{4}$$

Once the access policy is defined in the contract, the validator publishes it to the network.

$$Publish(ABAC) \rightarrow Set_{\text{Contract}} \tag{5}$$

To gain read/write access to the global ledger, a user initiates an access request.

$$ID_{\text{User}} \rightarrow AccessRequest_{\text{read/write}} \tag{6}$$

The global ledger invokes the access point ledger for that particular contract after receiving the request.

$$AccessRequest = \begin{cases} 0, & \text{Denied} \\ 1, & \text{Granted} \end{cases} \tag{7}$$

If the request is authorized, the user has access to the global ledger and can read and write to it. If it fails, the user will receive an error status message.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

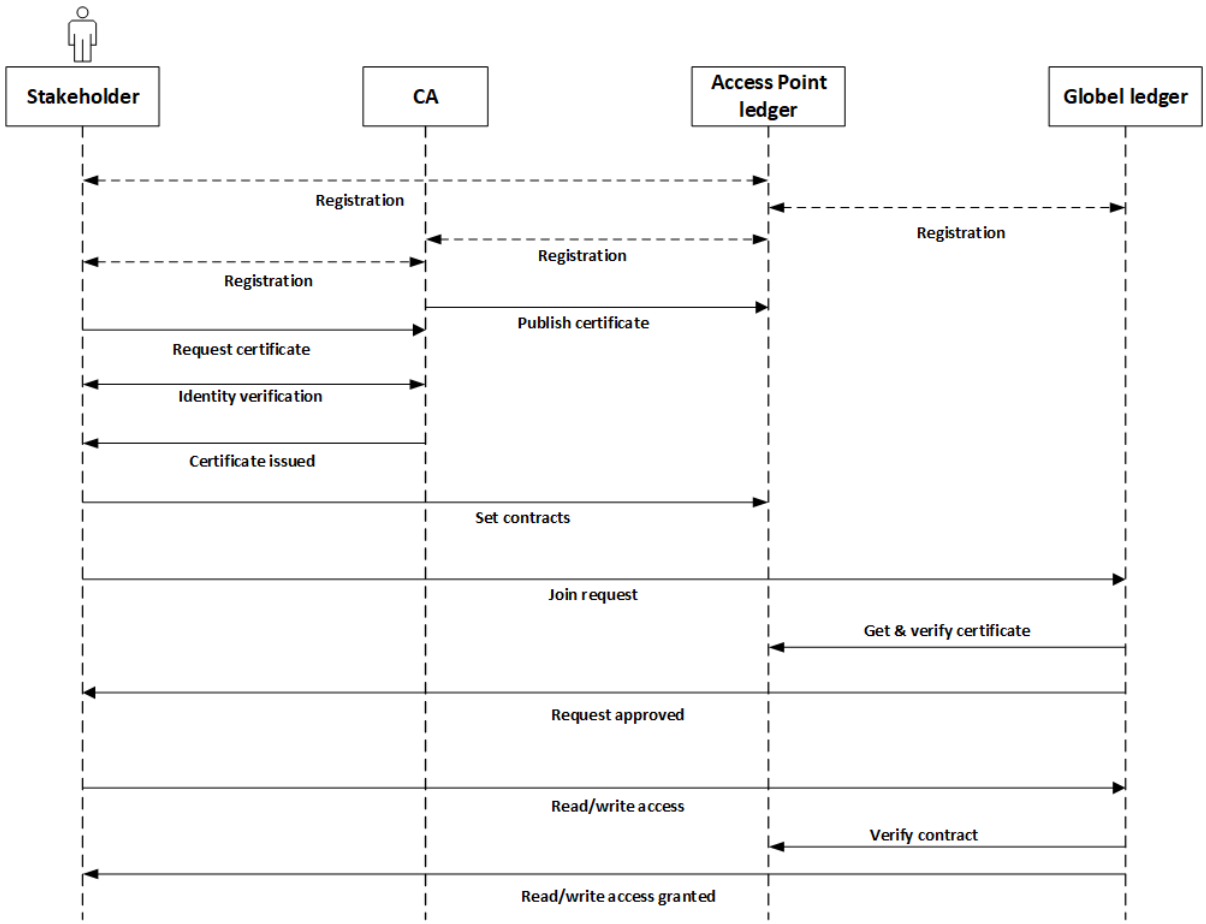

Figure 3: Workflow of *AccessChain*.

4.6. Primary functions of the AccessChain

Multiple functions can be provided by the framework to help with access control. These functions primarily consist of registering, updating, and deleting a business contract, as well as adding, updating, and deleting a business contract's access control policy. The following is a breakdown of how each function works.

i **Registering a new business contract** A number of businesses might agree on a new business deal, which is then registered by the contract's owner (i.e., the manufacturer) via the steps below.

- Create a contract for the new supply network (defining access privileges for each user).
- Register the contract on their local access point ledger and pay a fee to deploy the newly created contract onto the blockchain.

Notice that in our framework, any stakeholder can create a business contract as long as they have a legitimate business deal, for example, a group of retailers wants to enter into a short-term contract with local stores to reduce shortages.

ii **Updating an existing business contract** A number of businesses might agree on upgrading an existing contract, which is done by the contract's owner using the steps below.

- Create a new contract to replace the existing one.
- Register the contract to the local access point ledger and pay a cost to replace the existing contract on the blockchain with the newly formed one.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

iii **Deleting an existing business contract** Businesses can agree to terminate an existing contract that is due to expire but that they do not want to renew. The contract's owner can send a transaction to the local ledger to delete the existing contract's details.

iv **Adding and updating an access control policy** Businesses can agree to add an access control policy to a newly deployed contract, which is done by the contract's owner by sending a request to the contact's *policyAdd* service. Similarly, the owner can submit a request to call the contract's *policyUpdate* function to update an existing access control policy for a specific contract. However, our framework does not allow stakeholders to delete the contract's access control policy. The owner can only add or update user access privileges; if the owner tries to delete the access policy, the contract will immediately be nullified.

5. Experiment and Results

This section analyses performance statistics to demonstrate the feasibility of the proposed approach. The suggested framework was developed using JDK 15 in the Visual Studio Code experimental setting on a Windows 10 machine with an Intel Core i7 CPU operating at 2.21 GHz with 16 GB of RAM.

5.1. Performance Evaluation

The following three performance metrics were considered to validate the effectiveness of our approach:

i Time cost trade-off
ii Latency and Throughput
iii Scalability

To compare our approach to an alternative, we formed a second framework called "Uniform ledger", that does not need local ledgers for access request verification. We construct an ABAC access model for the uniform ledger, which is identical to *AccessChain*. Uniform ledger does not have separate ledgers for policies and contracts; rather, rather everything is accomplished in a single uniform ledger. The framework configuration is as follows:

- Read (get) the ledger and write the ledgers(add,update).
- Various network configurations with node counts ranging from 10 to 200.
- The workload varies between 50 and 1000 access requests in each network configuration.

5.1.1. Time cost trade-off

Time cost indicates the amount of time it takes to evaluate the access request and process it. The access response time $T(_{\mathrm{AR}})$ is calculated by using the following equation:

$$T(_{\mathrm{AR}}) = \sum_{i=0}^{n} (ARP_{\mathrm{t}} - AR_{\mathrm{t}}) \tag{8}$$

where n is the number of access requests, $\mathrm{ARP_t}$ is the time to process the request, $\mathrm{AR_t}$ is the access request time.

Figure 4 depicts the average response time for the total number of access requests for *AccessChain* and uniform ledger depending on different numbers of access requests. The request for access might be for either reading or writing. Once the frequency of access requests rises, the access response time significantly increases for our approach over the uniform ledger. This is driven by the fact that the *AccessChain* involves the access model and contracts. When an access request is made, the global validators looks for the contract that matches the request and responds, prolonging the time it takes for an access response. We consider this is appropriate, given the significant additional tasks required, such as validating permissions in local ledgers.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

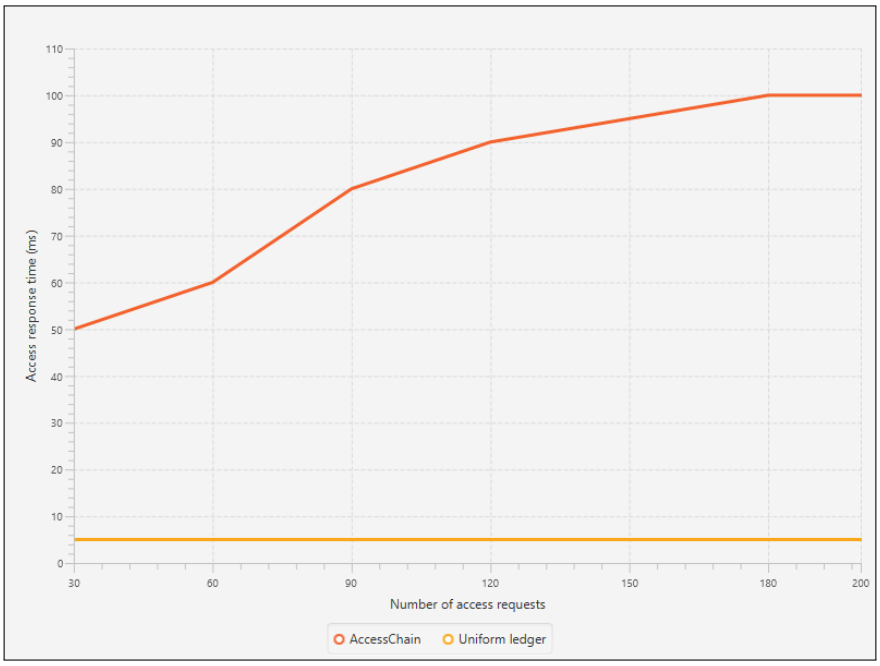

Figure 4: Comparison of time cost of Access response for different numbers of requests.

To examine the framework's accessibility further, we calculate the time required to complete various read and write requests, as shown in Figure 5. To begin, we send ten concurrent requests to *AccessChain*, then add a further ten concurrent requests at the same time, and continue the procedure until 100 concurrent requests are sent. We can see from the graph that write requests take longer than read requests.

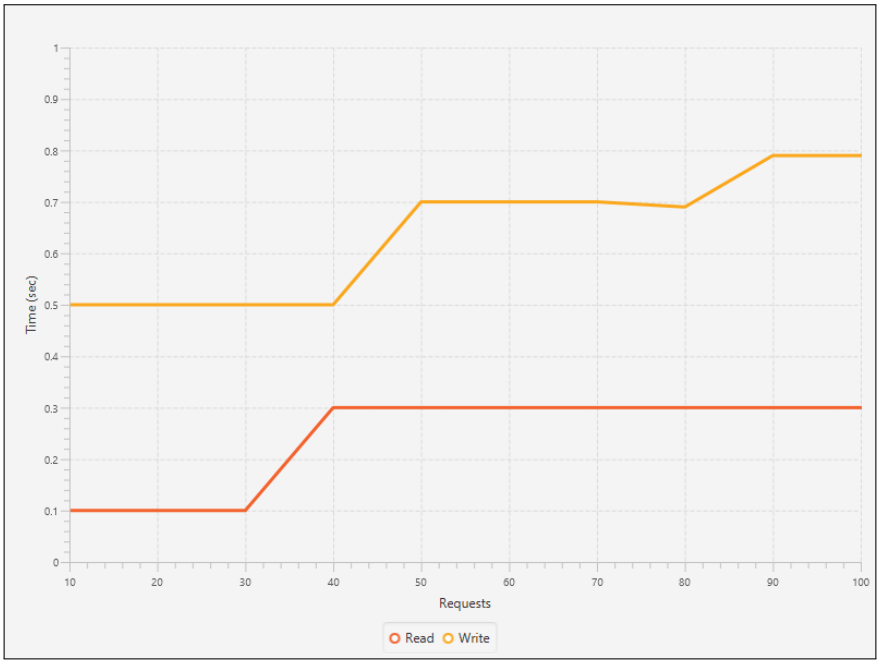

Figure 5: The trend of average cost time of *AccessChain* Read/Write operation at different numbers of requests

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

The standard deviations for the read and write transactions are shown in Table 2. It can be observed in the read transaction that each ledger's response time is quite similar. Therefore, the standard deviation is modest. It indicates that the read transactions (RT) are more consistent across the ledgers. The RT of each ledger in the write transaction has significant variability, indicating that the standard deviation is high. It implies that the write transaction is less consistent during the course of the experiment.

Transactions	Local L-01	Local L-02	Local L-03	Local L-04	Average response time	Std deviation
Read requests	100	200	400	1000	0.425	0.349
Write requests	50	190	300	1000	0.385	0.365

Table 2: The average and standard deviation of read/write requests.

5.1.2. Latency and Throughput

The throughput and latency performance measures are presented in this subsection. For throughput computations, the number of access requests is set between 50 and 1000. Figure 6 depicts the throughput of *AccessChain* and uniform ledger. As each stakeholder is identified on the local ledger and engaged on the global ledger, the graph depicts the throughput for both ledgers' complete transactions. The throughput improves linearly until a transaction send rate of 180 tps for 650 requests is reached, at which point it begins to drop. Even with 1000 requests, the framework maintains a throughput of around 140 tps. However, the uniform ledger maintains 114 tps for 1000 access requests.

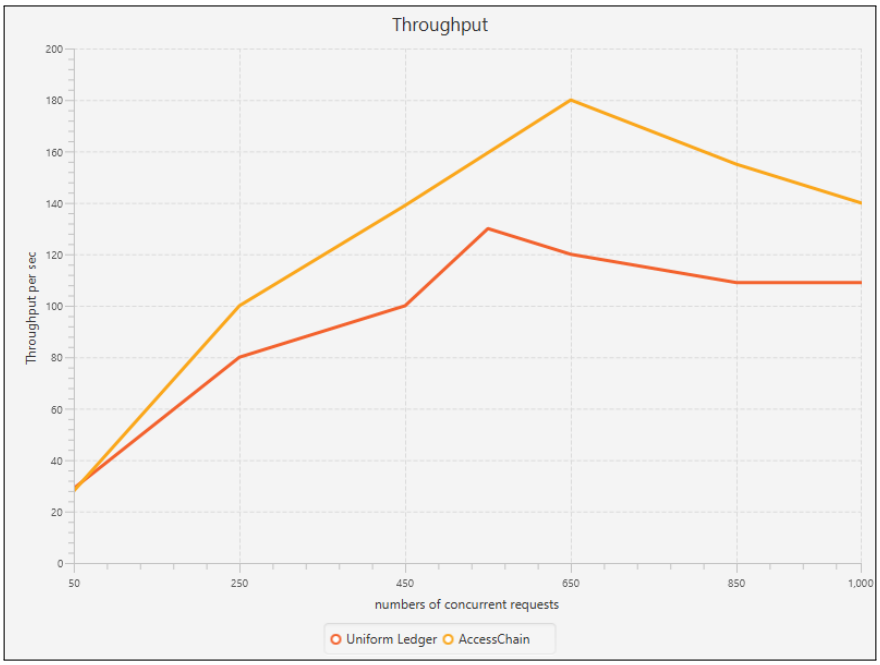

Figure 6: Throughput comparison of *AccessChain* with uniform ledger

We conducted 1000 requests in this series of trials to assess the framework's efficiency by evaluating the time it takes to finish different access requests. Access to the global ledger can be requested for reading or writing. Figure 7 depicts the outcomes of this experiment. As the number of access requests grows, the time it takes to complete the request grows linearly until it reaches 29 milliseconds for 650 requests. It reveals that the delay for processing 1000 requests is around 36 milliseconds which is quite acceptable. Besides, the graph displays the latency of a uniform

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

ledger with a similar amount of access requests. The uniform ledger has a delay of about 50 milliseconds.
Figures 7 and 6 show that given 1000 concurrent requests, throughput grows linearly to around 180 tps, and afterwards the framework goes beyond this threshold, resulting in lower throughput and higher latency. The results show that *AccessChain* can sustain excellent throughput in large-scale request settings.

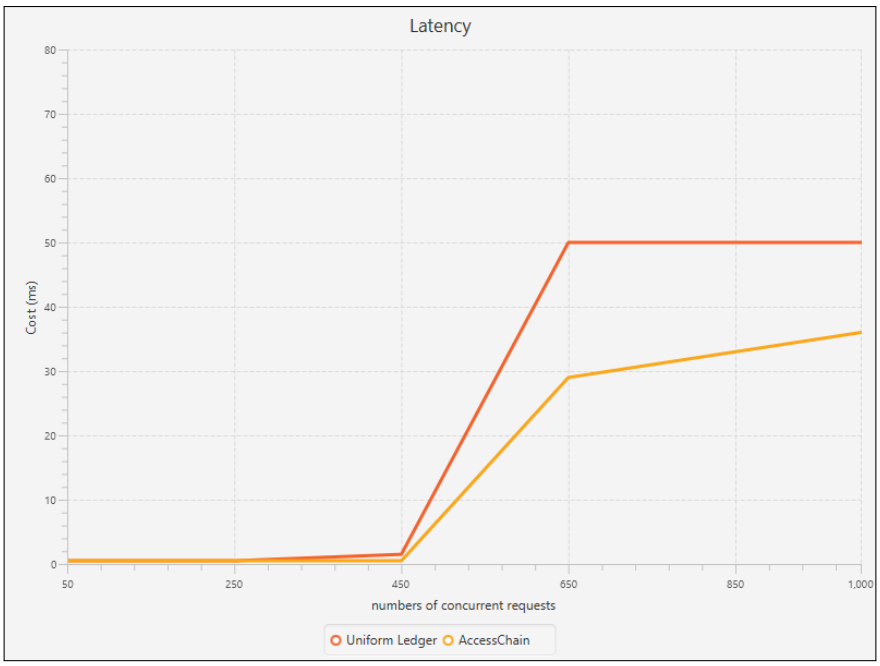

Figure 7: Latency comparison of *AccessChain* with uniform ledger.

5.1.3. Scalability

The objective of this set of experiments is to see how scalable the network is in terms of managing nodes. As the number of nodes grows in a network, the communication cost increases between ledgers. It also adds to the time it takes to verify and authenticate requests before allowing access to the global ledger. Figure 8 compares *AccessChain* to a uniform ledger when the network scales up. The average throughput for both frameworks declines as the number of nodes increases. However, the average throughput of *AccessChain* with 200 nodes rises to 624 transactions per second. On the other hand, uniform ledger fails to provide scalability, since throughput decreases significantly as the network grows up. To give more insight, Figure 9 provides the standard deviation (STDev) of the response value. In the context of performance analysis, the STDev of a transaction indicates whether it is stable throughout a sample or otherwise. Where a smaller STDev indicates that all iterations of the same transaction have similar response times (RT). So when the transaction amount decreases, the transaction RT become closer and the transaction becomes more consistent. Therefore, we may define it as a metric that shows how response times fluctuate around the mean.

$$Std(x_n) = \sqrt{\frac{n\sum x^2 - (\sum x^2)}{n(n-1)}} \tag{9}$$

Where the time is represented by x, and n represents the response magnitude.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

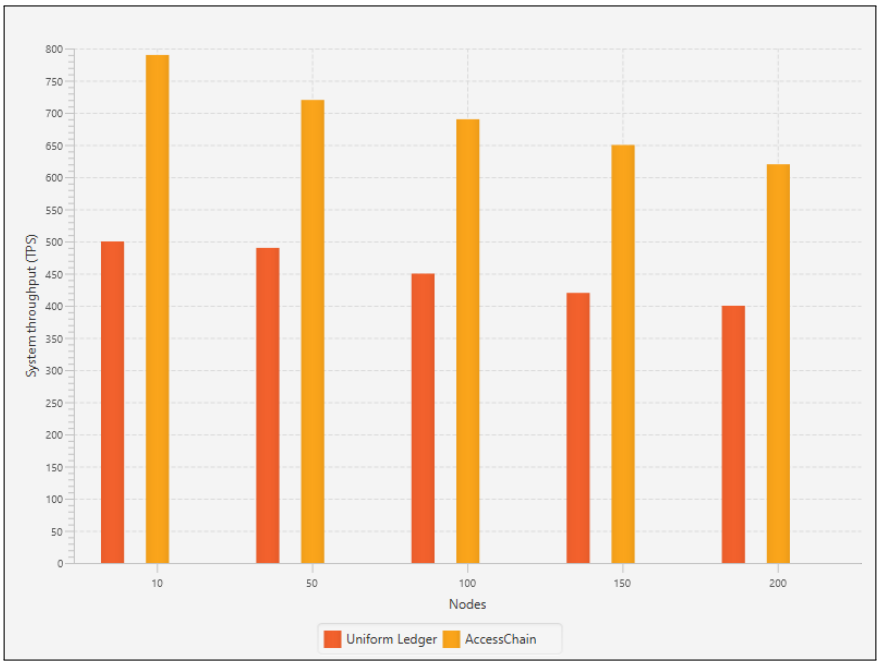

Figure 8: Comparison of framework throughput according to number of nodes

Figure 9 presents the response time with the four concurrent transaction amounts of 200, 500, 800, and 1000. The columns show the RT of four local ledgers. We can observe that when the number of local ledger grows, the RT grows linearly. However, when switching from one to two local ledgers, the rate of growth is faster than when switching from two to three and three to four local ledgers. In the case of having one local ledger, the average RT is 145 and the standard deviation is 7.8, the RT was in the range of 141 to 149 67% of the time. When compared to other numbers of local ledgers, this is rather consistent. However, as we increase the number of local ledgers, we can observe that RT has a wide range. It's worth noting that RT is just 65.73 milliseconds, even with 1000 transactions and four local ledgers. These statistics indicates that increasing the number of local ledgers might raise RT, implying that scalability can be accomplished at a minimal cost.

Table 3 represents the statics from 15 runs of this experiment.

Number of local ledgers	Average	Std deviation
Local L-01	145	7.83
Local L-02	157	31.63
Local L-03	240	45.31
Local L-04	182	65.73

Table 3: The average and standard deviation.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

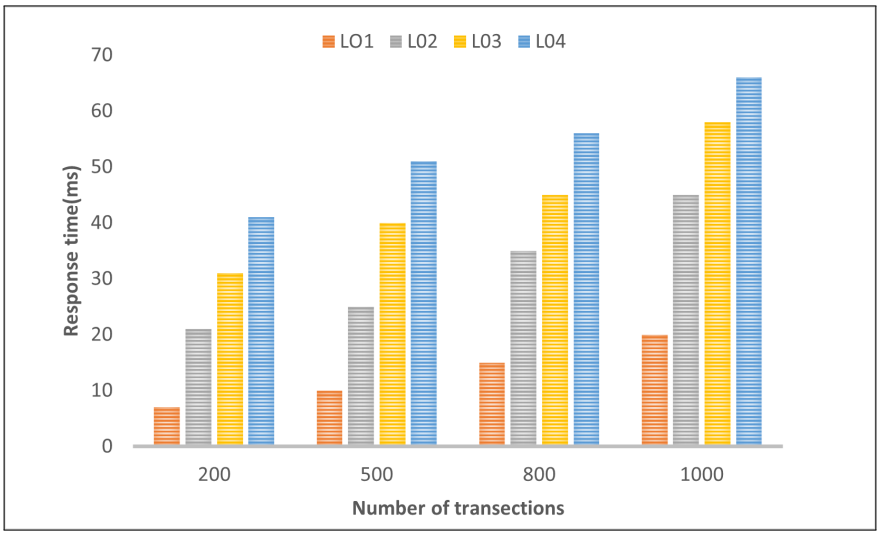

Figure 9: Response time vs number of transactions

5.2. *Security and Privacy Analysis*

We emphasize data privacy in *AccessChain* by implementing the following security measures. All data on the blockchain is encrypted with private-public key pairs. The data can only be decrypted by the user who has the corresponding private key and granted access permissions. This subsection summarizes our threat model and examines the critical security risks, along with *AccessChain's* resiliency over threats.

framework Threat model In this section, we look at *AccessChain*'s potential threats. Given that stakeholders can place their contracts on a public ledger and trade on a private ledger, some dishonest traders may take advantage of this by putting additional contracts on the public ledger to get access to read/write unauthorized transactions. The adversary may be able to view or change user information. We presume that the access contract maintained on the access point ledger is only valid for a limited time and for a limited set of permissions granted to participants. We also assume that data stored on the global ledger is encrypted and unavailable to anyone without the decryption key because *AccessChain's* primary goal is to secure sensitive data. In our threat model, we investigate how dishonest traders might carry out specific destructive acts in order to disrupt the *AccessChain* framework.

i Creating multiple Contracts: An adversary may create several contracts but never use them in trading. The network load may be increased if there are a significant number of business contracts on the local ledger. To combat this, all contracts include a deposit, similar to how most public ledgers demand a minimum fee amount before a transaction is allowed. Moreover, in local ledgers, each trader has a contract creation threshold for a defined time period.

ii Breaching Access contract: A trader may seek access to a contract and then attempt to access another contract after it has been provided. In such cases, the accessValidation function is responsible since it acts as a watchdog, ensuring that any access provided for a certain contract expires after a set period of time and must be regenerated. Also, the signed cipher of access issued by the local validator assures that the provided access is only valid for the specified contract. As a result, establishing an additional access without the validators' permission is impossible. Validator can revoke authorization if an adversary is identified attempting to access unauthorized data by updating the ledgers and eliminating any privileges associated with the adversary.

iii Local ledger Validaters: A local ledger validator performs most of the blockchain-based operations, such as access control and setting contracts. One may argue, that putting too much faith in a single administrative institution could risk the framework. A blockchain administrator may act maliciously, putting the framework's security at risk. Public blockchains become strongly secure when paired with Proof of Work (POW). Our local ledgers are public ledgers, and we made this decision while keeping our contracts and access regulations in mind, due to the way public blockchains work with POW. We argue that POW makes it extremely difficult to change any part of the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0106.v1

blockchain since it requires re-mining all subsequent blocks. It is difficult for a validator to dominate the network's processing power due to the high cost of the equipment and power required to execute the hash functions.

6. Conclusion

We developed a blockchain-based framework with ABAC model to assure data privacy by adopting a distributed framework to enable fine-grained, dynamic access control management for SCM. In order to solve the scalability issue, the framework helps by offering a two-tiered network design. A global ledger is used to record transactions, while access policies and business contracts are kept in multiple local ledgers. The framework enables a systematic approach that advantages the supply chain, and the experiments yield convincing results. Furthermore, the threat model depicts how resilient our framework is against a wide range of threats. Furthermore, the results of the performance monitoring show that *AccessChain's* response time with four local ledgers is acceptable, and therefore it provides significantly greater scalability.
For future work, we aim to add more local ledgers to evaluate the framework's scalability and throughput. Secondly, the existing approach could be optimized by using high-performance hash techniques to boost data processing efficiency.

Author Contributions:
Aaliya Sarfaraz: Conceptualisation, Investigation, Methodology, Resources, Visualisation, Software, Writing-original draft.
Ripon K. Chakrabortty: Conceptualisation, Investigation, Methodology, Resources, Visualisation, Validation, Supervision, Writing- review editing.
Daryl L. Essam: Conceptualisation, Investigation, Methodology, Resources, Visualisation, Validation, Supervision, Writing- review editing

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.

References

[1] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Decentralized Business Review (2008) 21260.
[2] A. Sarfaraz, R. K. Chakrabortty, D. L. Essam, A blockchain-coordinated supply chain to minimize bullwhip effect with an enhanced trust consensus algorithm (2021).
[3] A. Sarfaraz, R. K. Chakrabortty, D. L. Essam, A tree structure-based improved blockchain framework for a secure online bidding system, Computers & Security 102 (2021) 102147.
[4] M. Berneis, H. Winkler, Value proposition assessment of blockchain technology for luxury, food, and healthcare supply chains, Logistics 5 (2021) 85.
[5] A. Dorri, S. S. Kanhere, R. Jurdak, Towards an optimized blockchain for iot, in: 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), IEEE, 2017, pp. 173–178.
[6] M. Pilkington, Blockchain technology: principles and applications, in: Research handbook on digital transformations, Edward Elgar Publishing, 2016.
[7] T. H. Yuen, Pachain: Private, authenticated and auditable consortium blockchain, in: International Conference on Cryptology and Network Security, Springer, 2019, pp. 214–234.
[8] A. Pinna, W. Ruttenberg, Distributed ledger technologies in securities post-trading revolution or evolution?, ECB Occasional Paper (2016).
[9] O. Tene, J. Polonetsky, Privacy in the age of big data: a time for big decisions, Stan. L. Rev. Online 64 (2011) 63.
[10] D. Bechtsis, N. Tsolakis, E. Iakovou, D. Vlachos, Data-driven secure, resilient and sustainable supply chains: gaps, opportunities, and a new generalised data sharing and data monetisation framework, International Journal of Production Research (2021) 1–21.
[11] C. Reimsbach-Kounatze, Enhancing access to and sharing of data: Striking the balance between openness and control over data, in: Data Access, Consumer Interests and Public Welfare, Nomos Verlagsgesellschaft mbH & Co. KG, 2021, pp. 25–68.
[12] Y. Hong, J. Vaidya, S. Wang, A survey of privacy-aware supply chain collaboration: From theory to applications, Journal of Information Systems 28 (2014) 243–268.
[13] T. Uesugi, Y. Shijo, M. Murata, Design and evaluation of a privacy-preserving supply chain system based on public permissionless blockchain, in: 2021 International Symposium on Electrical, Electronics and Information Engineering, 2021, pp. 312–321.
[14] X. Gao, Z. Xiang, H. Wang, J. Shen, J. Huang, S. Song, An approach to security and privacy of rfid system for supply chain, in: IEEE international conference on e-commerce technology for dynamic e-business, IEEE, 2004, pp. 164–168.
[15] F. Tian, An information system for food safety monitoring in supply chains based on HACCP, blockchain and internet of things, Ph.D. thesis, WU Vienna University of Economics and Business, 2018.
[16] T. Ferdousi, D. Gruenbacher, C. M. Scoglio, A permissioned distributed ledger for the us beef cattle supply chain, IEEE Access 8 (2020) 154833–154847.

[17] H. Wu, J. Cao, Y. Yang, C. L. Tung, S. Jiang, B. Tang, Y. Liu, X. Wang, Y. Deng, Data management in supply chain using blockchain: Challenges and a case study, in: 2019 28th International Conference on Computer Communication and Networks (ICCCN), IEEE, 2019, pp. 1–8.
[18] S. Algarni, F. Eassa, K. Almarhabi, A. Almalaise, E. Albassam, K. Alsubhi, M. Yamin, Blockchain-based secured access control in an iot system, Applied Sciences 11 (2021) 1772.
[19] I. Yaqoob, K. Salah, R. Jayaraman, Y. Al-Hammadi, Blockchain for healthcare data management: Opportunities, challenges, and future recommendations, Neural Computing and Applications (2021) 1–16.
[20] J. Lai, F. Guo, W. Susilo, X. Huang, P. Jiang, F. Zhang, Data access control in cloud computing: Flexible and receiver extendable, IEEE Transactions on Services Computing (2021).
[21] A. Ouaddah, H. Mousannif, A. Abou Elkalam, A. A. Ouahman, Access control in the internet of things: Big challenges and new opportunities, Computer Networks 112 (2017) 237–262.
[22] P. Samarati, S. C. de Vimercati, Access control: Policies, models, and mechanisms, in: International School on Foundations of Security Analysis and Design, Springer, 2000, pp. 137–196.
[23] R. Xu, Y. Chen, E. Blasch, G. Chen, Blendcac: A blockchain-enabled decentralized capability-based access control for iots, in: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 2018, pp. 1027–1034.
[24] X. Jin, R. Krishnan, R. Sandhu, A unified attribute-based access control model covering dac, mac and rbac, in: IFIP Annual Conference on Data and Applications Security and Privacy, Springer, 2012, pp. 41–55.
[25] U. Khalid, M. Asim, T. Baker, P. C. Hung, M. A. Tariq, L. Rafferty, A decentralized lightweight blockchain-based authentication mechanism for iot systems, Cluster Computing (2020) 1–21.
[26] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, et al., Spanner: Google's globally distributed database, ACM Transactions on Computer Systems (TOCS) 31 (2013) 1–22.
[27] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, B. Ford, Omniledger: A secure, scale-out, decentralized ledger via sharding, in: 2018 IEEE Symposium on Security and Privacy (SP), IEEE, 2018, pp. 583–598.
[28] M. Zamani, M. Movahedi, M. Raykova, Rapidchain: A fast blockchain protocol via full sharding., IACR Cryptol. ePrint Arch. 2018 (2018) 460.
[29] J. S. Park, J. Hwang, Role-based access control for collaborative enterprise in peer-to-peer computing environments, in: Proceedings of the eighth ACM symposium on Access control models and technologies, 2003, pp. 93–99.
[30] L. Wang, D. Wijesekera, S. Jajodia, A logic-based framework for attribute based access control, in: Proceedings of the 2004 ACM workshop on Formal methods in security engineering, 2004, pp. 45–55.
[31] J. Park, R. Sandhu, The uconabc usage control model, ACM transactions on information and system security (TISSEC) 7 (2004) 128–174.
[32] S. Gusmeroli, S. Piccione, D. Rotondi, A capability-based security approach to manage access control in the internet of things, Mathematical and Computer Modelling 58 (2013) 1189–1205.
[33] R. S. Sandhu, P. Samarati, Access control: principle and practice, IEEE communications magazine 32 (1994) 40–48.
[34] G. D. Skinner, et al., Cyber security management of access controls in digital ecosystems and distributed environments, in: 6th International Conference on Information Technology and Applications (ICITA 2009), 2009, pp. 77–82.
[35] L. Gong, et al., A secure identity-based capability system., in: IEEE symposium on security and privacy, 1989, pp. 56–63.
[36] B. Anggorojati, P. N. Mahalle, N. R. Prasad, R. Prasad, Capability-based access control delegation model on the federated iot network, in: The 15th International Symposium on Wireless Personal Multimedia Communications, IEEE, 2012, pp. 604–608.
[37] J. L. Hernández-Ramos, A. J. Jara, L. Marin, A. F. Skarmeta, Distributed capability-based access control for the internet of things, Journal of Internet Services and Information Security (JISIS) 3 (2013) 1–16.
[38] S. Pal, T. Rabehaja, A. Hill, M. Hitchens, V. Varadharajan, On the integration of blockchain to the internet of things for enabling access right delegation, IEEE Internet of Things Journal 7 (2019) 2630–2639.
[39] Y. Wang, M. Singgih, J. Wang, M. Rit, Making sense of blockchain technology: How will it transform supply chains?, International Journal of Production Economics 211 (2019) 221–236.
[40] S. A. Abeyratne, R. P. Monfared, Blockchain ready manufacturing supply chain using distributed ledger, International Journal of Research in Engineering and Technology 5 (2016) 1–10.
[41] O. J. A. Pinno, A. R. A. Gregio, L. C. De Bona, Controlchain: Blockchain as a central enabler for access control authorizations in the iot, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, IEEE, 2017, pp. 1–6.
[42] Y. Zhu, Y. Qin, Z. Zhou, X. Song, G. Liu, W. C.-C. Chu, Digital asset management with distributed permission over blockchain and attribute-based access control, in: 2018 IEEE International Conference on Services Computing (SCC), IEEE, 2018, pp. 193–200.
[43] D. D. F. Maesa, P. Mori, L. Ricci, Blockchain based access control, in: IFIP international conference on distributed applications and interoperable systems, Springer, 2017, pp. 206–220.
[44] A. Ouaddah, A. Abou Elkalam, A. Ait Ouahman, Fairaccess: a new blockchain-based access control framework for the internet of things, Security and communication networks 9 (2016) 5943–5964.
[45] A. Ouaddah, A. Abou Elkalam, A. A. Ouahman, Towards a novel privacy-preserving access control model based on blockchain technology in iot, in: Europe and MENA cooperation advances in information and communication technologies, Springer, 2017, pp. 523–533.
[46] L. Xu, L. Chen, N. Shah, Z. Gao, Y. Lu, W. Shi, Dl-bac: Distributed ledger based access control for web applications, in: Proceedings of the 26th International Conference on World Wide Web Companion, 2017, pp. 1445–1450.
[47] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, J. Wan, Smart contract-based access control for the internet of things, IEEE Internet of Things Journal 6 (2018) 1594–1605.
[48] H. Liu, D. Han, D. Li, Fabric-iot: A blockchain-based access control system in iot, IEEE Access 8 (2020) 18207–18218.
[49] G. Zyskind, O. Nathan, et al., Decentralizing privacy: Using blockchain to protect personal data, in: 2015 IEEE Security and Privacy Workshops, IEEE, 2015, pp. 180–184.
[50] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, J. He, Blochie: a blockchain-based platform for healthcare information exchange, in: 2018 ieee

 doi:10.20944/preprints202202.0106.v1

international conference on smart computing (smartcomp), IEEE, 2018, pp. 49–56.
[51] J. Flapper, User access control on the blockchain for supply chain visibility, B.S. thesis, University of Twente, 2019.
[52] O. Novo, Blockchain meets iot: An architecture for scalable access management in iot, IEEE Internet of Things Journal 5 (2018) 1184–1195.
[53] K. R. K. Reddy, A. Gunasekaran, P. Kalpana, V. R. Sreedharan, S. A. Kumar, Developing a blockchain framework for the automotive supply chain: A systematic review, Computers & Industrial Engineering 157 (2021) 107334.
[54] K. Singh, Blockchain technology: A potential game changer for automotive industry., International Journal of Advanced Research in Management and Social Sciences 9 (2020) 49–55.
[55] L. M. Powell, J. Schwartz, M. Hendon, The mobility open blockchain initiative: Identity, members, technologies, and future trends, in: Revolutionary Applications of Blockchain-Enabled Privacy and Access Control, IGI Global, 2021, pp. 99–118.
[56] N. Madenas, A. Tiwari, C. Turner, S. Peachey, An analysis of supply chain issues relating to information flow during the automotive product development, Journal of Manufacturing Technology Management (2015).
[57] V. C. Hu, D. Ferraiolo, D. R. Kuhn, et al., Assessment of access control systems, Citeseer, 2006.
[58] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, J. Voas, Attribute-based access control, Computer 48 (2015) 85–88.
[59] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, et al., Guide to attribute based access control (abac) definition and considerations (draft), NIST special publication 800 (2013) 1–54.
[60] Q.-h. Bai, Y. Zheng, Study on the access control model, in: Proceedings of 2011 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference, volume 1, IEEE, 2011, pp. 830–834.
[61] E. Coyne, T. R. Weil, Abac and rbac: scalable, flexible, and auditable access management, IT professional 15 (2013) 14–16.
[62] S. Bhatt, T. K. Pham, M. Gupta, J. Benson, J. Park, R. Sandhu, Attribute-based access control for aws internet of things and secure industries of the future, IEEE Access 9 (2021) 107200–107223.
[63] H. Kim, D.-K. Kim, A. Alaerjan, Abac-based security model for dds, IEEE Transactions on Dependable and Secure Computing (2021).
[64] E. Yuan, J. Tong, Attributed based access control (abac) for web services, in: IEEE International Conference on Web Services (ICWS'05), IEEE, 2005.
[65] D. Hankerson, A. J. Menezes, S. Vanstone, Guide to elliptic curve cryptography, Springer Science & Business Media, 2006.
[66] V. B. Pavel Khahulin Igor Barinov, Poa network white paper, Sep. 2018. URL: `https://github.com/poanetwork/wiki/wiki/POA-Network-Whitepaper`.